

Listing of Claims:

1. (Currently Amended) A method of handling editing operations of objects on a video display in an application running on a computer system comprising:

providing an application program interface with a text data model ~~having one or more~~ abstraction layer[[s]], wherein the text data model abstraction layer provides access to a text store and is invoked by a text editing layer for performing ~~having a text editing operations~~ functionality;

detecting an edit operation for an object displayed on the video display by the computer system; and

sending an edit operation request to ~~one of the~~ non-abstract[[ion]] layer[[s]] via the application program interface to initiate editing of the object [[via]] by invoking the text data model abstraction layer.

2. (Currently Amended) The method of handling editing operations in claim 1 wherein the application program interface further comprises ~~has one or more a text view model~~ abstraction layer[[s]] that is invoked by a text layout layer for performing ~~having a layout editing operations~~ functionality.

3. (Currently Amended) The method of handling editing operations in claim 1 wherein the ~~application program interface~~ text data model comprises[[has]] a text container abstract[[ion]] class[[layer]] for storing a linear piece of text and a text position abstract[[ion]] class[[layer]] for identifying a location within the piece of text.

4. (Currently Amended) The method of handling editing operations in claim 3 further comprising a text navigator abstract[[ion]] class[[layer]] for moving between one or more locations within a[[the]] text container.

5. (Currently amended) A system for editing objects displayed by a computer comprising:

a processor; and

a memory coupled with and readable by the processor and containing instructions that, when executed by the processor, cause the processor to detect an edit operation for an object displayed on [[the]] a video display by the computer system, send an edit operation request from an application program via an application program interface to an abstraction layer within the interface to initiate editing of the object by the abstraction layer causing the abstraction layer to receive the edit operation request, determine a container type for a container in which the object is displayed, read a set of properties related to the object to be edited, read a set of properties related to the container in which the object is displayed to determine a type for the container, and edit the object based on the container type and the received edit operation request.

6. (Original) The system according to claim 5 wherein the type of container is a text container having a framework class.

7. (Original) The system according to claim 6 wherein the framework class includes a dependency object class used as an abstract representative for text structuring elements.

8. (Original) The system according to claim 7 wherein the framework class includes a dependency property object class containing formatting information on the dependency object class.

9. (Original) The system according to claim 5 further comprising one or more base types.

10. (Original) The system according to claim 9 wherein one of the base types is a logical direction type for indicating direction in linear text space.

11. (Original) The system according to claim 6 further comprising an abstract text position class defining a mechanism for identifying location within text in the text container.

12. (Original) The system according to claim 6 further comprising an abstract class text navigator providing a content exploration functionality within the object.

13. (Original) The system according to claim 6 further comprising a view model for the text container to provide one or more presentational characteristics of the text container.

14. (Currently amended) A ~~machine-readable~~ computer storage medium encoding a computer program of instructions for executing a computer process of editing objects displayed by a computer system, said computer process comprising:

providing an application program interface with a text data model having ~~one or more~~ abstraction layer[[s]], wherein the text data model abstraction layer provides access to a text store and is invoked by a text editing layer for performing ~~having a text editing operations~~ functionality;

detecting an edit operation for an object displayed by the computer system; and
sending an edit operation request to ~~one of the~~ non-abstract[[ion]] layer[[s]] via the application program interface to initiate editing of the object [[via]] by invoking the text data model abstraction layer.

15. (Currently Amended) The computer storage medium of claim 14 wherein the application program interface further comprises ~~has one or more~~ a text view model abstraction layer[[s]] that is invoked by a text layout layer for performing ~~having a layout editing operations~~ functionality.

16. (Currently Amended) The computer storage medium of claim 15 wherein the application program interface comprises[[has]] a text container abstract[[ion]] class[[layer]] for storing a linear piece of text and a text position abstract[[ion]] class[[layer]] for identifying a location within the piece of text.

17. (Currently Amended) The computer storage medium of claim 16 further comprising the interface having a text navigator abstract[[ion]] class[[layer]] for moving between one or more locations within the text container.